
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Minoru Oka, et al

Attorney Docket No.: OMRNP081

Application No.: 10/799,930

Examiner: C.J. Barnes

Filed: March 12, 2004

Group: 2121

Title: CONTROL SYSTEM AND METHOD
FOR ON-LINE EDITING OF USER PROGRAM



CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail to: Commissioner for Patents, Alexandria, Virginia 22313 on August 29, 2005.

Signed: _____

Deborah Neill

TRANSMITTAL OF CERTIFIED COPY OF PRIORITY DOCUMENT

Commissioner for Patents
Alexandria, Virginia 22313

Sir:

Enclosed herewith are certified copies of priority documents Japan patent application No. 2004-059865 filed on March 3, 2004, and Japan patent application No. 2003-068920 filed on March 13, 2003. Please file these documents in the subject application.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Kenichi Nishimura
Registration No. 29,093

P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 3 月 1 3 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 0 6 8 9 2 0
Application Number:
[ST. 10/C] : [J . P 2 0 0 3 - 0 6 8 9 2 0]

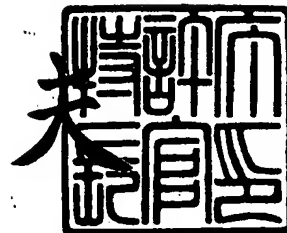
出 願 人 オムロン株式会社
Applicant(s):

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 4 年 3 月 2 4 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 61717

【提出日】 平成15年 3月13日

【あて先】 特許庁長官殿

【国際特許分類】 G05B 19/05

【発明者】

 【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
 番地 オムロン株式会社内

 【氏名】 岡 実

【発明者】

 【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
 番地 オムロン株式会社内

 【氏名】 出来 仁太郎

【発明者】

 【住所又は居所】 京都府京都市下京区西洞院木津屋橋通東入ル オムロン
 ソフトウェア株式会社内

 【氏名】 矢尾板 宏心

【発明者】

 【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
 番地 オムロン株式会社内

 【氏名】 市村 勝彦

【発明者】

 【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
 番地 オムロン株式会社内

 【氏名】 小野 彰男

【特許出願人】

 【識別番号】 000002945

 【氏名又は名称】 オムロン株式会社

 【代表者】 立石 義雄

【代理人】

【識別番号】 100092598

【弁理士】

【氏名又は名称】 松井 伸一

【手数料の表示】

【予納台帳番号】 019068

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9800459

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 CPUユニット及びユーザプログラム編集方法

【特許請求の範囲】

【請求項 1】 ユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおける CPUユニットにおいて、

前記ユーザプログラムと同一のユーザプログラムを格納するテンポラリユーザメモリを設け、

前記ユーザメモリに対する編集処理を行う際には、前記テンポラリユーザメモリに格納された前記ユーザプログラムを実行するように構成したことを特徴とする CPUユニット。

【請求項 2】 ユーザプログラムを演算実行するプログラマブルコントローラにおける CPUユニットにおいて、

同一の前記ユーザプログラムを格納する複数のユーザメモリを備え、

前記複数のユーザメモリのいずれか一方を動作中にするるとともに、他方を待機中にする手段と、

前記待機中のユーザメモリに格納されたユーザプログラムが編集された場合に、その待機中のユーザメモリを動作中に切り替えるとともに、動作中になったユーザメモリに格納された編集後のユーザプログラムを待機中になったユーザメモリにコピーする機能を備えたことを特徴とする CPUユニット。

【請求項 3】 前記ユーザプログラム中の編集対象が、ファンクションブロックであることを特徴とする請求項 1 または 2 に記載の CPUユニット。

【請求項 4】 ユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおける CPUユニットにおけるユーザプログラム編集方法において、

前記 CPUユニットには、前記ユーザプログラムと同一のユーザプログラムを格納するテンポラリユーザメモリを設け、

前記ユーザメモリに対する編集処理を行うに先立ち、前記テンポラリユーザメモリに格納された前記ユーザプログラムに基づいて演算実行を継続し、

その状態で前記ユーザメモリに格納された前記ユーザメモリを修正し、

修正後、前記ユーザメモリに格納された編集後のユーザプログラムに基づいて演算実行をするようにしたことを特徴とするユーザプログラム編集方法。

【請求項 5】 同一のユーザプログラムを格納する複数のユーザメモリを備え、一方のユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおける CPU ユニットのユーザプログラム編集方法において、

前記複数のユーザメモリのいずれか一方を動作中にするとともに、他方を待機中に設定し、

ツールからの命令に従い前記待機中のユーザメモリに格納されたユーザプログラムを編集し、

その編集が完了後、その待機中のユーザメモリを動作中に切り替え、

次いで、CPU ユニットは、動作中に切り替わったユーザメモリに格納された編集後のユーザプログラムを、待機中になったユーザメモリにコピーすることを特徴とするユーザプログラム編集方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

この発明は、CPU ユニット及びユーザプログラム編集方法に関するものである。

【0 0 0 2】

【従来の技術】

ファクトリーオートメーション（FA）の制御装置として、プログラマブルコントローラ（PLC）が用いられている。この PLC は、複数のユニットから構成される。すなわち、電源供給源の電源ユニット、PLC 全体の制御を統率する CPU ユニット、FA の生産装置や設備装置の適所に取り付けしたスイッチやセンサの信号を入力する入力ユニット、アクチュエータなどに制御出力を出す出力ユニット、通信ネットワークに接続するための通信ユニットなどの各種のユニットを適宜組み合わせて構成される。

【0 0 0 3】

PLCのCPUユニットにおける制御は、入力ユニットで入力した信号をCPUユニットのI/Oメモリに取り込み（INリフレッシュ）、予め登録されたユーザプログラム記述言語（例えばラダー言語）で組まれたユーザプログラムに基づき論理演算をし（演算実行）、その演算実行結果をI/Oメモリに書き込んで出力ユニットに送り出し（OUTリフレッシュ）、その後、いわゆる周辺処理を行うということをサイクリックに繰り返し処理するようになる。

【0004】

そして、上記したユーザプログラムは、CPUユニットのユーザメモリに格納される。さらに、ユーザプログラムの構成要素には、複数の回路を組み合わせて部品化したファンクションブロックもある。

【0005】

ところで、実際に稼動した後における不具合の発生、その他の理由から、稼動中のユーザプログラムに対して、追加・削除・変更などの編集を行う必要がある。係る修正処理の一態様として、FAシステムの制御を実行しつつ、プログラムの編集を行うオンラインエディットがある。

【0006】

このオンラインエディットは、編集対象のプログラム部分の直前にジャンプ命令等を一時的に追加し、上記のサイクリックな演算実行する際には編集対象部分を飛ばして実行することにより、システムを停止することなくサイクリックな処理を継続できる。この状態で、ユーザプログラムの編集対象部分に対し追加・削除・変更などの修正を行うようにするものである。

【0007】

【発明が解決しようとする課題】

しかしながら、上記した従来のオンラインエディットは、1回の処理で1つの回路に対して行うことが多く、ユーザプログラムの複数箇所に点在する複数回路を一度に修正するのは困難であった。そして、編集後は、ジャンプ命令等を削除する必要があり、係る処理が煩雑であるばかりでなく、削除し忘れると編集したプログラムが機能しない。

【0008】

また、複数回路から構成されるファンクションブロックに対するオンラインエディットも従来実行できなかった。すなわち、従来の通常の回路に対する編集の方法をファンクションブロックに適用すると、大量の追加・削除が一度に発生してしまう（ファンクションブロックのインスタンスの追加，削除）ため、ユーザプログラムの実行性能に大きな影響を与える。

【0009】

さらに、ファンクションブロックの場合、どのファンクションブロックが使用されるかを特定するファンクションブロックレコード記録領域と、実際のプログラム部分の両方に対して編集処理を行うとともに、実際のプログラム部分では、入力パラメータと出力パラメータ並びに実際の演算を行うボディ部分というように複数箇所を同時に編集する必要があるので、編集中のプログラム管理が煩雑となる。

【0010】

さらに、編集対象部分はスキップされるので、制御に必要不可欠なプログラム部分に対して編集処理すると、システム全体に悪影響を与えるおそれがある。従って、必ずしもすべてのプログラム部分に対してオンラインエディットが行えるとは限らない。

【0011】

この発明は、オンラインエディットによるプログラムの編集・修正が容易に行うことができ、かつ、編集対象（ファンクションブロックか否か）並びに個数に関係なく編集処理が行えるCPUユニット及びユーザプログラム編集方法を提供することを目的とする。

【0012】

【課題を解決するための手段】

この発明によるCPUユニットは、ユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおけるCPUユニットである。そして、前記ユーザプログラムと同一のユーザプログラムを格納するテンポラリユーザメモリを設け、前記ユーザメモリに対する編集処理を行う際には、前記テンポラリユーザメモリに格納された前記ユーザプログラムを実行するように構成

した。

【0013】

また、別の解決手段としては、ユーザプログラムを演算実行するプログラマブルコントローラにおけるCPUユニットにおいて、同一の前記ユーザプログラムを格納する複数のユーザメモリを備え、前記複数のユーザメモリのいずれか一方を動作中にするとともに、他方を待機中にする手段と、前記待機中のユーザメモリに格納されたユーザプログラムが編集された場合に、その待機中のユーザメモリを動作中に切り替えるとともに、動作中になったユーザメモリに格納された編集後のユーザプログラムを待機中になったユーザメモリにコピーする機能を備えることである。

そして、いずれの場合も、前記ユーザプログラム中の編集対象が、ファンクションブロックでも良いし、通常のプログラムでも良い。

【0014】

一方、本発明に係るユーザプログラム編集方法は、ユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおけるCPUユニットにおけるユーザプログラム編集方法において、前記CPUユニットには、前記ユーザプログラムと同一のユーザプログラムを格納するテンポラリユーザメモリを設け、前記ユーザメモリに対する編集処理を行うに先立ち、前記テンポラリユーザメモリに格納された前記ユーザプログラムに基づいて演算実行を継続し、その状態で前記ユーザメモリに格納された前記ユーザメモリを修正し、修正後、前記ユーザメモリに格納された編集後のユーザプログラムに基づいて演算実行をするようにした。

【0015】

また、本発明に係るユーザプログラム編集方法の別の解決手段としては、同一のユーザプログラムを格納する複数のユーザメモリを備え、一方のユーザメモリに格納されたユーザプログラムを演算実行するプログラマブルコントローラにおけるCPUユニットのユーザプログラム編集方法において、前記複数のユーザメモリのいずれか一方を動作中にするとともに、他方を待機中に設定し、ツールからの命令に従い前記待機中のユーザメモリに格納されたユーザプログラムを編

集し、その編集が完了後、その待機中のユーザメモリを動作中に切り替え、次いで、CPUユニットは、動作中に切り替わったユーザメモリに格納された編集後のユーザプログラムを、待機中になったユーザメモリにコピーするようにした。

【0016】

この発明によれば、ユーザプログラムを格納するユーザメモリを複数用意し、その複数のユーザメモリに同一のユーザプログラムを格納した。なお、このように同一のユーザプログラムを格納する方法としては、ツールを用いて一方のユーザメモリに格納後、コピー機能によりその一方のユーザメモリに格納されたユーザプログラムを他方のユーザメモリに格納するようにしても良いし、各ユーザメモリに対してそれぞれツールを用いてダウンロードして格納するようにしても良い。

【0017】

格納されたユーザプログラムに対して編集をする場合、複数のユーザメモリには、同一内容のユーザプログラムが格納されているので、一方のユーザメモリに格納されたユーザプログラムに対して編集処理をしている間、他方のユーザメモリに格納されたユーザプログラムに基づいて演算実行することができる。これにより、オンラインエディットが行える。しかも、同一内容のプログラムであり、オンラインエディット中に動作するユーザプログラムに対しては編集処理をしていないので、何ら制限無く演算実行が行える。

【0018】

また、編集対象のユーザプログラムが格納されたユーザメモリは、待機中となり、編集処理中は実際の制御に使用されないので、複数箇所を処理することもできるし、ファンクションブロックのように大量の追加削除が一度に発生してしまうようなプログラム要素に対する編集も問題なくできる。さらに、一時的にジャンプ命令などを書き込む必要も無く、作業が簡便化される。

【0019】

そして、編集が完了したならば、今まで編集対象で待機中になっていたユーザメモリを動作中（実行中）に切替え、今まで動作中であったユーザメモリを待機中にするにより、編集された新しいユーザプログラムに基づく演算処理が行

える。

【0020】

【発明の実施の形態】

図1は、本発明の好適な一実施の形態を実現するシステム全体を示している。CPUユニット10は、PLCの構成要素の一つである。そして、システムプログラムは、システムROM11に格納されており、そのシステムプログラムは、MPU12に呼び出され、そのMPU12において、ワークメモリとしてのシステムRAM13を適宜使用しながら、プログラムに従った所定の処理を実行する。また、運転時において、ユーザデータの1つであるユーザプログラムは、ユーザメモリ14に格納されている。さらに、I/Oデータやパラメータは、IOMメモリ15に格納されている。

【0021】

そして、これらユーザメモリ14に格納されたユーザプログラム（命令オブジェクトコード）は、命令実行エンジン（ASIC）16に順次呼び出され、そこにおいて呼び出した命令オブジェクト（可逆展開可能なレベルの中間コード（ニモニックコード等））を解析し、命令実行をする。また、この命令実行時に、適宜IOMメモリ15にアクセスし、I/Oデータを読み書きしたり、パラメータを取得する。さらに、I/Oデータは、システムバス17を介して他のユニットひいては、マスタユニット経由でスレーブとの間で送受される。さらに、通信インタフェース18を備え、ツール20との間でデータの送受を行えるようになっている。

【0022】

また、本実施の形態のPLCは、ファンクションブロック（FB）に対応している。ファンクションブロックは、図2（a）に示すように、引数とも呼ばれる入力パラメータ（図示の例では「D0，D1」）と、戻り値とも呼ばれる出力パラメータ（図示の例では「D2，H1.0」）と、実際に演算実行する処理を規定するボディBとを備える。そして、ボディBの中身は、例えば図2（b）に示すようになる。ラダー言語で記述されたユーザプログラム中では、ファンクションブロックは、図2（a）に示すようにボディ部分はブラックボックスのように

箱を記述し、実際の演算実行に際し、ファンクションブロックが呼び出されると、図 2 (b) に示すような複数の回路からなるプログラムを実行する。

【0023】

つまり、図 3 に示すように、ユーザプログラムを上から順番に実行していき (ST1)、ファンクションブロックの呼出し命令を受けると (ST2)、インスタンス前処理として、入出力パラメータの値をファンクションブロック実行領域にコピーする (ST3)。次いで、FB ボディ処理として、ファンクションブロック実行領域を参照してボディに記述されたラダープログラム (図 2 (b) 等) を実行する (ST4)。その後、インスタンス後処理として、ファンクションブロック実行領域から出力パラメータにコピーし、ファンクションブロック部分の処理を終了する (ST5)。次いで、ファンクションブロックに続くユーザプログラムのメインルーチンを実行する (ST6)。なお、係るファンクションブロックの実行処理手順自体は、従来公知のものであるので、その詳細な説明を省略する。

【0024】

ここで、本発明では、通常のユーザメモリ (表ユーザメモリ) 14 に加え、テンポラリユーザメモリ (裏ユーザメモリ) 19 を設けた。ユーザメモリ 14 は、通常動作時のユーザプログラムを格納する領域であり、運転実行時はこのユーザメモリ 14 からユーザプログラムを呼び出して実行する。また、ツール 20 からのアクセス許可領域は、ユーザメモリ 14 であり、テンポラリユーザメモリ 19 に対してはアクセスできないようになっている。

【0025】

また、テンポラリユーザメモリ 19 は、正常実行時のユーザメモリ 14 に格納されたユーザプログラムと同一のプログラムが格納されており、オンラインエディット中には、ユーザメモリ 14 に変わってテンポラリユーザメモリ 19 に格納されたユーザプログラムを実行する。これにより、オンラインエディット中も、オンラインエディット前のユーザプログラムと同一内容のユーザプログラムに基づいた演算処理をすることができる。

【0026】

そして、このように従前と同一のユーザプログラムを実行中に、このオンラインエディットによりユーザメモリ 1 4 に格納したユーザプログラムの編集が行える。なお、ユーザメモリ 1 4 と、テンポラリユーザメモリ 1 9 は、物理的に異なるメモリ（記憶素子）を用いても良いし、同一のメモリを用い、メモリ割付で 2 つのメモリ領域を設けても良い。

【0 0 2 7】

上記した処理を行うための M P U 1 2 の具体的な処理は、以下のようにになっている。まず、ユーザプログラムのダウンロードは、図 4 に示すフローチャートを実施するようになっている。すなわち、ツール 2 0 からのダウンロードを受けると、取得したユーザプログラムをユーザメモリ 1 4 に書き込む（S T 1 1：図 5（a）参照）。

【0 0 2 8】

ダウンロード終了後、サムチェックなどにより正常にダウンロードできたか否かを判断する（S T 1 2）。そして、正常でない場合には、ステップ 1 1 に戻り再度ダウンロードを受ける。なお、正常にダウンロードできたか否かの判断結果を、ツール 2 0 へ通知する機能も M P U 1 2 は持っている。

【0 0 2 9】

また、正常にダウンロードできた場合には、そのユーザメモリ 1 4 にダウンロードしたユーザプログラムを、テンポラリユーザメモリ 1 9 にもコピーして記憶保持させる（S T 1 3：図 5（b）参照）。

【0 0 3 0】

次に、ユーザプログラムの編集処理アルゴリズムについて説明する。以下の説明では、ファンクションブロックの追加を例にあげて説明しているが、ファンクションブロックの削除や、内容の変更など各種のものに適用できる。さらには、編集対象のプログラムは、ファンクションブロックに限ることはなく、通常のプログラムでもよい。そして、同時に編集対象となるプログラム数も 1 個でも良いし複数個でも良い。特に本発明が有為な効果を奏するのは、ファンクションブロックのときであり、次に、通常のプログラムを複数箇所編集する場合がくる。

【0 0 3 1】

そして、ファンクションブロックを追加する処理を実行する場合には、図 6 に示すフローチャートを実行する。すなわち、まず、プログラム実行対象を、テンポラリユーザメモリ 1 9 に切り替える（S T 2 1）。この切り替えは、例えば、現在の実行プログラムを格納するユーザメモリ 1 4（テンポラリメモリ 1 9）を特定するフラグを設け、そのフラグを書き換えることにより、実現できる。そして、このように切り替えることにより、命令実行エンジン 1 6 に対しては、テンポラリユーザメモリ 1 9 に格納されたユーザプログラム（命令オブジェクト）が提供され、実際の制御が継続される。図 4 のステップ 1 3 を実行したことにより、ユーザメモリ 1 4 の内容とテンポラリユーザメモリ 1 9 の内容が一致しているため、今まで処理していたのと同じの制御を継続して行うことができる。

【 0 0 3 2 】

次に、ユーザメモリ 1 4 のファンクションブロックボディプログラム領域を確保する（S T 2 2：図 7（a）参照）。つまり、ユーザメモリ 1 4 のメモリマップは、図 8 に示すように、先頭にプログラムの基本情報（サム値など）を記録する領域があり、次に、使用されているファンクションブロックの情報を記憶する F b レコード部と、実際のユーザプログラムを格納するプログラムレコードがある。ユーザプログラム中では、該当する位置に呼び出しプログラムを記述するようになっている。そして、F B レコードのすぐ後ろにプログラムレコードが記述されるようになっている。

【 0 0 3 3 】

そこで、図 9 に示すように、所定位置に F B Z__4 を追加する場合、ユーザメモリ 1 4 は、F B レコードからプログラムレコードまで連続して形成されているため、F B Z が新規のものとする、F B レコードに F B Z についての情報を書き込む領域を確保する必要がある。そこで、その F B Z を挿入する位置よりも下側（アドレスの大きいもの）に対し、F B Z の分だけ移動させる。

【 0 0 3 4 】

次いで、確保した領域に対して、追加する F B ボディブロックをダウンロードして追加する（S T 2 3：図 7（b））。これにより、図 9（a）の状態から図 9（b）に示すように正常にダウンロードができたことになる。

【 0 0 3 5 】

次に、サム値チェックなどにより正常にダウンロードしたか否かを判断し（S T 2 4）、正常にダウンロードできたならば、ステップ 2 5 に飛ぶ。このとき、ユーザプログラムのメインルーチンの該当部分について、呼び出しプログラムを記述する（S T 2 5）。

【 0 0 3 6 】

そして、この呼び出しプログラムの追加が正常に完了すると、プログラム実行をユーザメモリへ戻す（S T 2 6：図 7（c）参照）。これに伴い、ユーザメモリ 1 4 に格納された編集後のユーザプログラムを実行することになり、より高精度な制御が行える。

【 0 0 3 7 】

次いで、ユーザメモリ 1 4 の内容をテンポラリユーザメモリ 1 9 にコピーする（S T 2 7：図 7（d）参照）。実際には、一度にユーザメモリの全体をコピーすることはできないので、P L C のサイクリックな処理中の周辺処理時に少しずつコピーすることになる。

【 0 0 3 8 】

なお、上記したユーザプログラム 1 4 へのダウンロード、追加、移動などの各処理や、ユーザメモリ 1 4 からテンポラリメモリ 1 9 へのコピー処理などは、実際には、M P U 1 2 からの指示を受けた命令実行エンジン 1 6 が行う。

【 0 0 3 9 】

また、上記した実施の形態では、編集処理の一態様である追加について説明したが、削除並びに編集も同様に行える。つまり、いずれの場合にも編集処理をするに先立ち、プログラム実行をテンポラリユーザメモリ 1 9 側に移す。これにより、ユーザメモリ 1 4 に対して編集処理しても、その処理が外部に影響を与えなくなる。そして、その状態でユーザメモリ 1 4 中の不要なファンクションブロックに関する情報を削除したり、ボディプログラムを編集したりすることができる。

【 0 0 4 0 】

なお、上記した実施の形態では、ファンクションブロックについて説明したが

、本発明はこれに限ることはなく、ファンクションブロックでない通常の回路に付いても編集が行える。

【0 0 4 1】

【発明の効果】

以上のように、この発明では、ユーザメモリに加えてテンポラリコピーを設けるなど、同一のユーザプログラムを備えた複数のユーザメモリを設けたため、あるユーザメモリに対して編集処理をしている場合には、別のユーザメモリに格納されたユーザプログラムに基づいて制御することができる。つまり、装置全体としてオンラインエディットを取りながら、修正対象のユーザプログラムは、待機中で実際の制御に使用されないため、自由に修正することができる。よって、オンラインエディットによるプログラムの修正が容易に行うことができ、かつ、編集対象（ファンクションブロックか否か）並びに個数に関係なく編集処理が行える。

【図面の簡単な説明】

【図 1】

本発明に係る CPU ユニットの一実施の形態を示すブロック図である。

【図 2】

ファンクションブロックを説明する図である。

【図 3】

ファンクションブロックを含むラダープログラムの実行処理を示すフローチャートである。

【図 4】

MPU の機能（ダウンロード）を示すフローチャートである。

【図 5】

MPU のダウンロードの作用を説明する図である。

【図 6】

MPU の機能（ファンクションブロックの追加）を示すフローチャートである。

。

【図 7】

M P U の機能（ファンクションブロックの追加）の作用を説明する図である。

【図 8】

ユーザメモリのメモリマップの一例を示す図である。

【図 9】

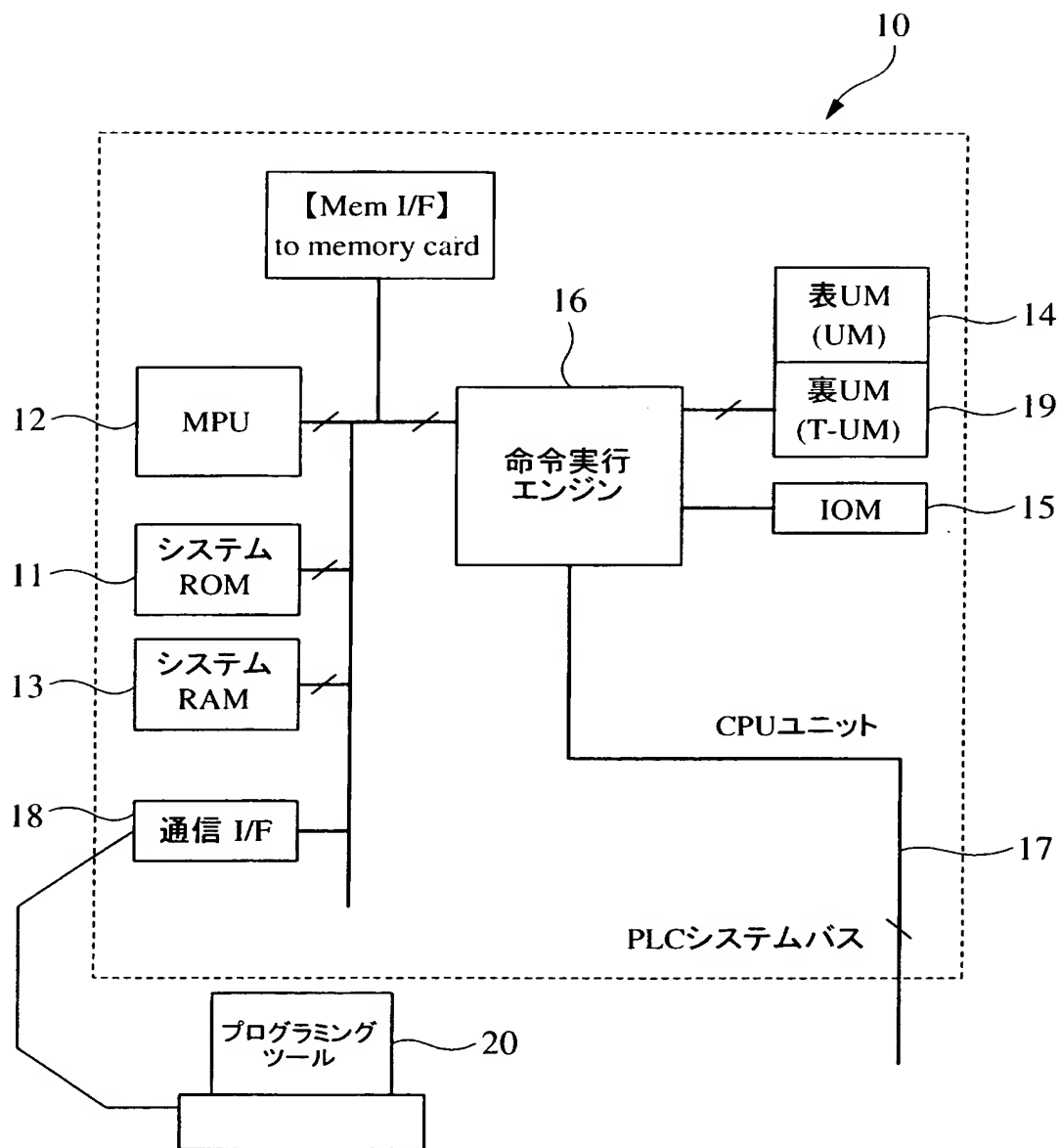
M P U の機能（ファンクションブロックの追加）の作用を説明する図である。

【符号の説明】

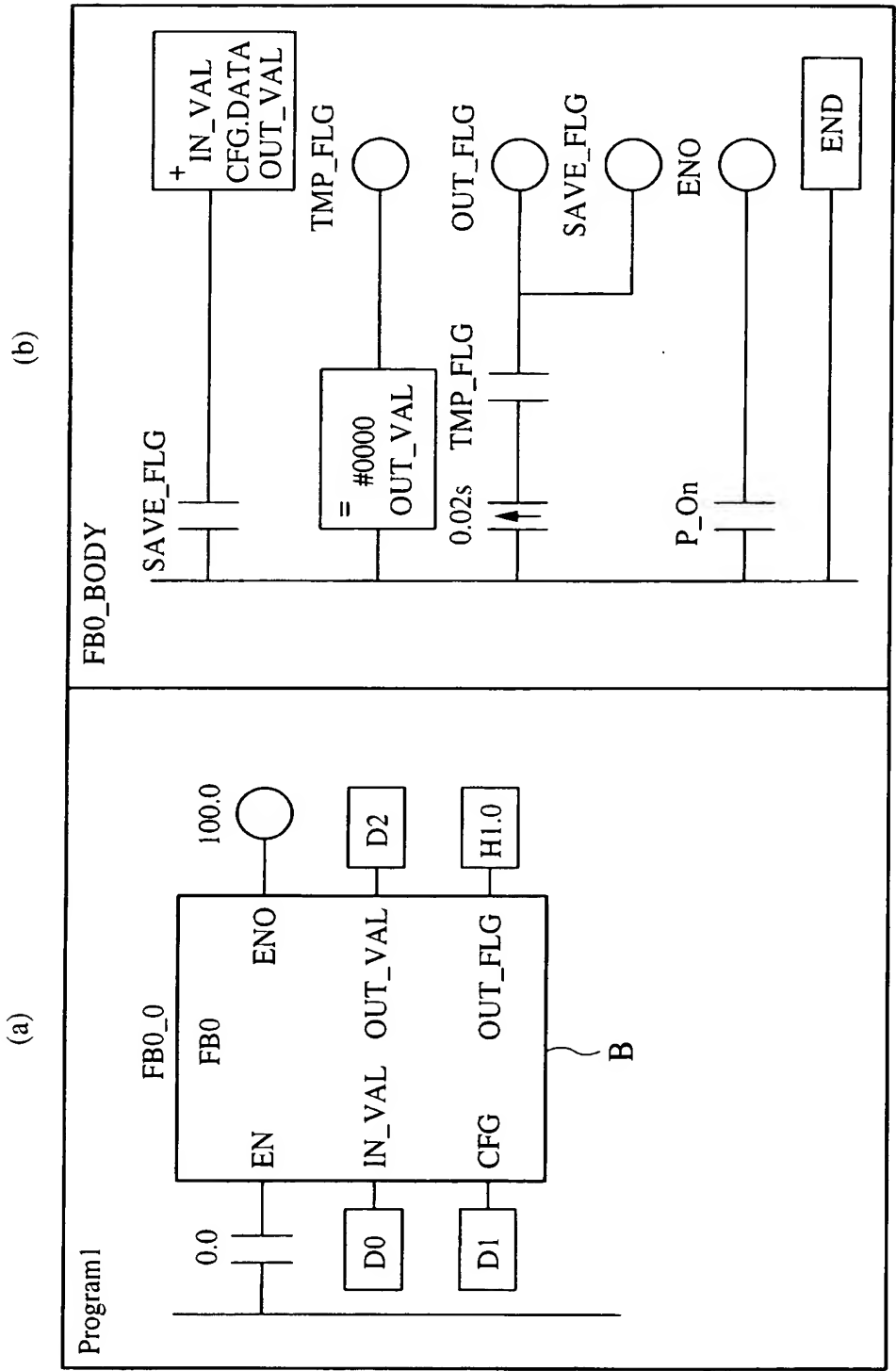
- 1 0 C P U ユニット
- 1 1 システム R O M
- 1 2 M P U
- 1 3 システム R A M
- 1 4 ユーザメモリ
- 1 5 I O メモリ
- 1 6 命令実行エンジン
- 1 7 システムバス
- 1 8 通信インタフェース
- 1 9 テンポラリユーザメモリ
- 2 0 ツール

【書類名】 図面

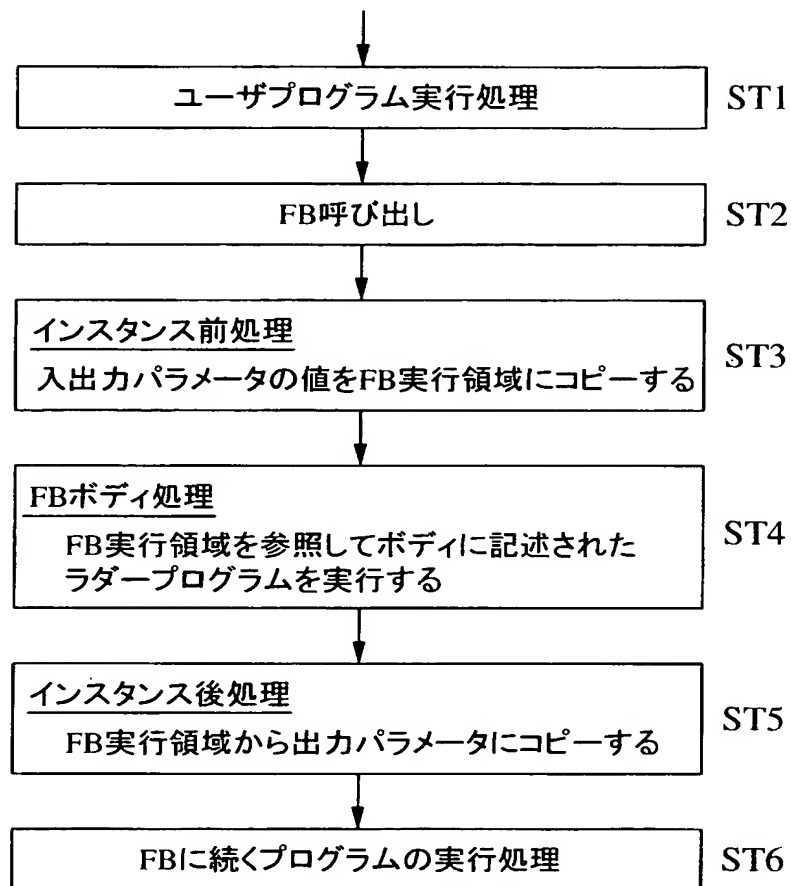
【図 1】



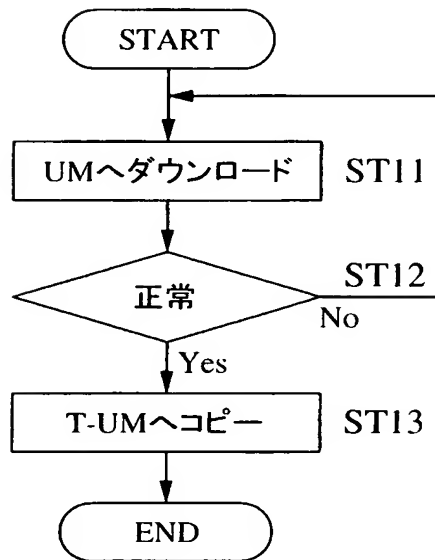
【図 2】



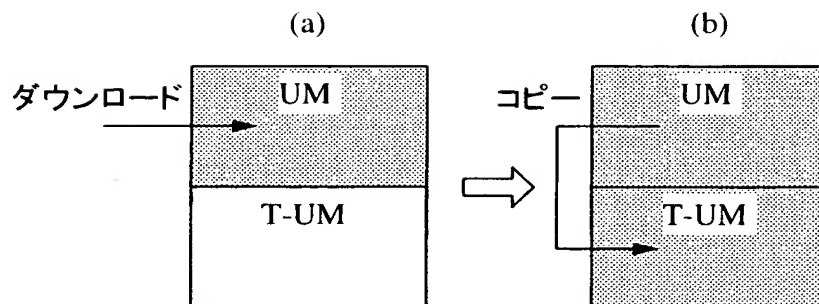
【図 3】



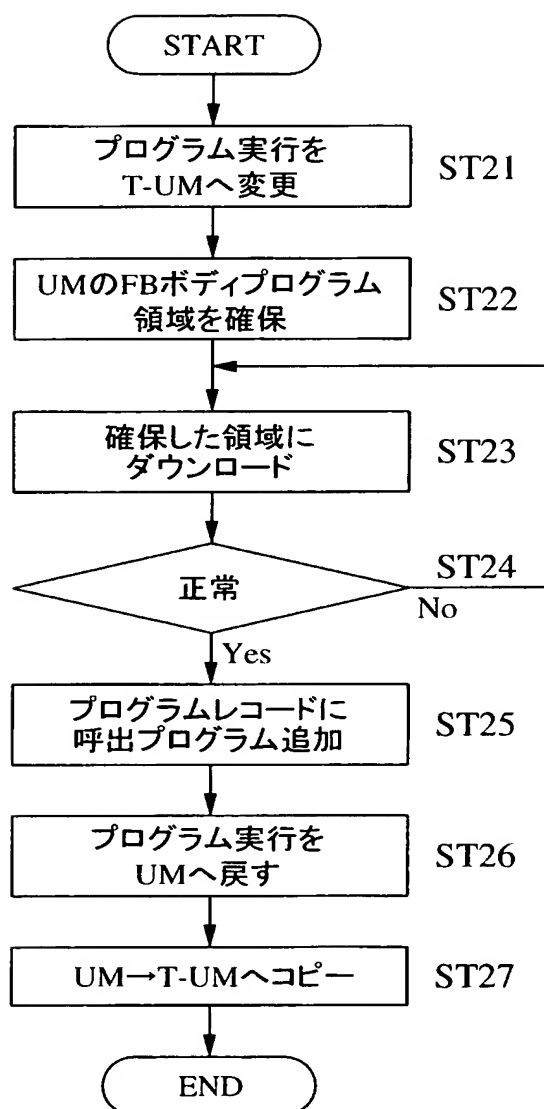
【図 4】



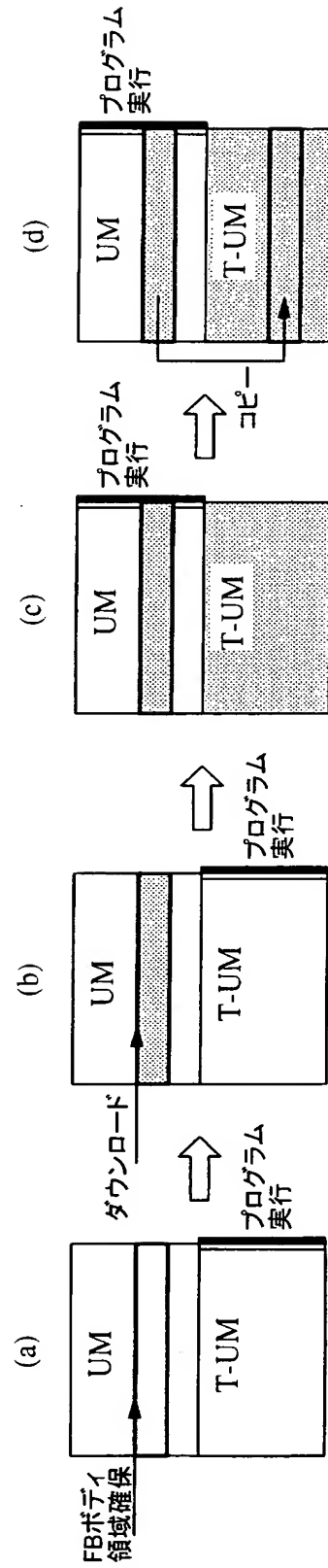
【図 5】



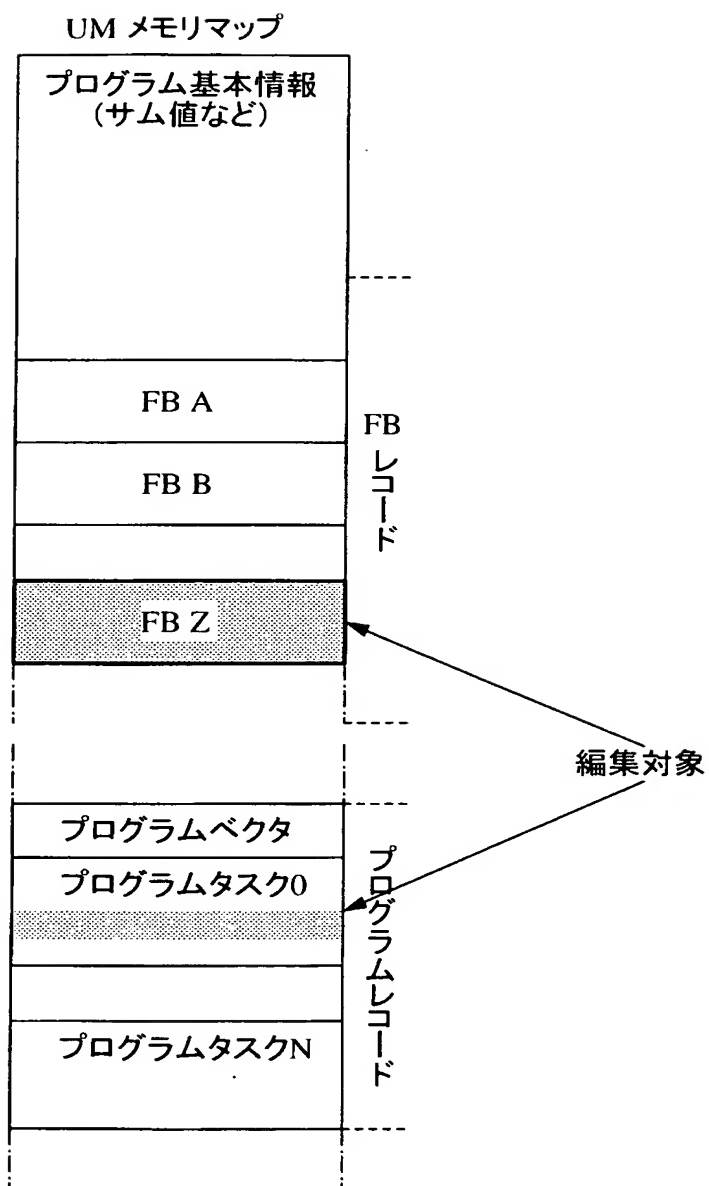
【図 6】



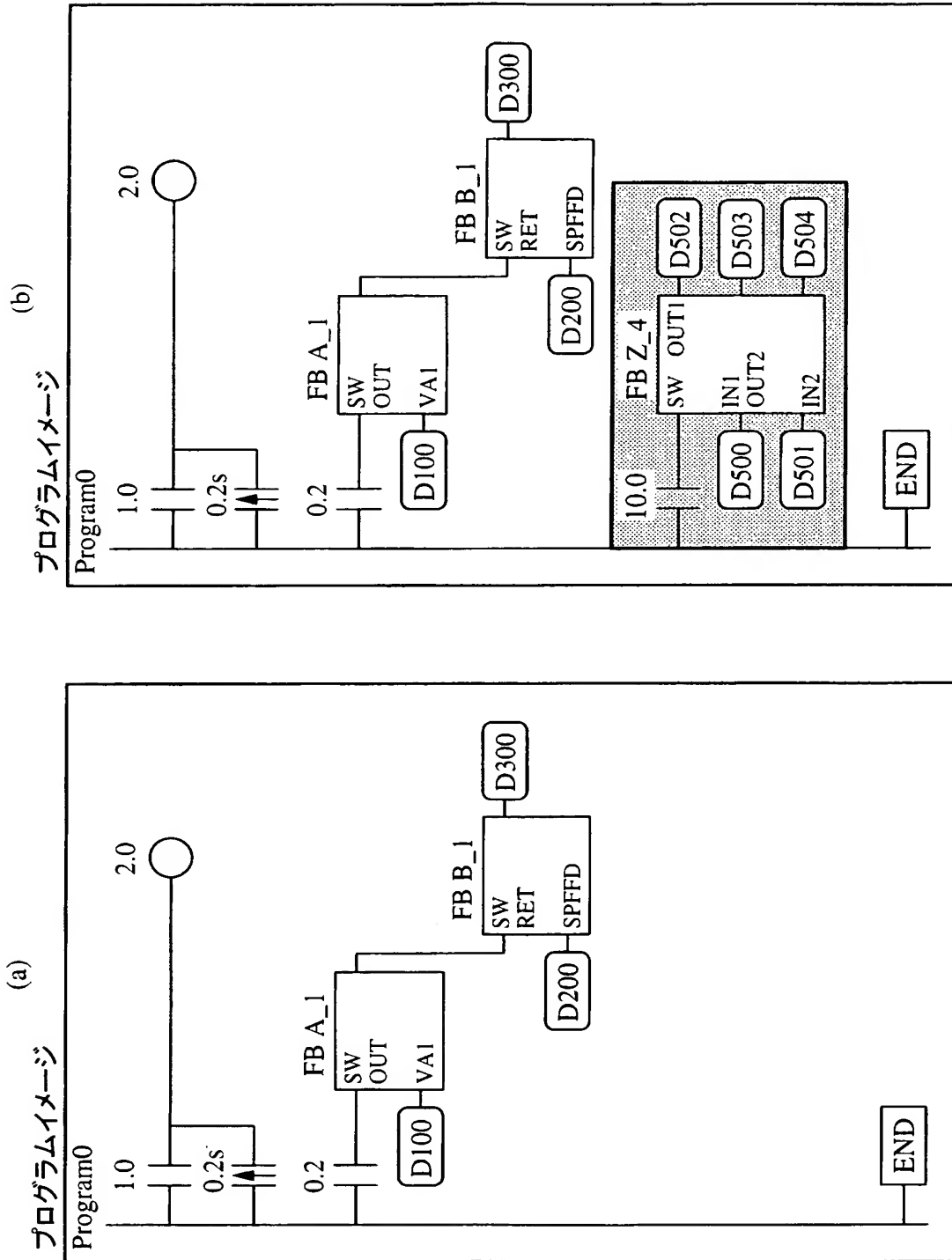
【図 7】



【図 8】



【図 9】



【書類名】 要約書

【要約】

【課題】 オンラインエディットによるプログラムの修正が容易に行うことができ、かつ、編集対象（ファンクションブロックか否か）並びに個数に関係なく編集処理が行える CPU ユニットを提供すること

【解決手段】 ユーザメモリ 14 に格納されたユーザプログラムを演算実行するプログラマブルコントローラにおける CPU ユニット 10 である。そして、ユーザプログラムと同一のユーザプログラムを格納するテンポラリユーザメモリ 19 を設け、ユーザメモリに対する編集処理を行う際には、テンポラリユーザメモリに格納されたユーザプログラムが命令実行エンジン 16 に呼び出され、演算実行される。編集後は、ユーザメモリに格納された編集済みユーザプログラムに基づき実行され、その編集済みユーザプログラムは、テンポラリユーザメモリにコピーされる。

【選択図】 図 1

特願 2 0 0 3 - 0 6 8 9 2 0

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 2 9 4 5]

1. 変更年月日	2 0 0 0 年 8 月 1 1 日
[変更理由]	住所変更
住 所	京都市下京区塩小路通堀川東入南不動堂町 8 0 1 番地
氏 名	オムロン株式会社